SYSTEM AND METHOD FOR PROVIDING REDUNDANT DATA LOAD SHARING IN A DISTRIBUTED NETWORK

TECHNICAL FIELD OF THE INVENTION

The present invention relates in general to data maintenance and more particularly to a system and method for providing redundant data load sharing in a distributed network.

BACKGROUND OF THE INVENTION

In update and query systems requiring real time response, the approach to guarantee the fastest access to data is to hold the data in physical memory. In situations where that data is crucial to the operation of the querying device, redundancy is also implemented such that a failure of a single hardware element storing this information does not prevent subsequent successful queries. This situation is compounded in systems that are highly distributed and where the storage of data is decentralized among peer devices.

Similar problems have been solved in a variety of ways. Many of these solutions rely on a master source for the stored data or at least an ability to re-fetch that data. Such is the case with the use of network caching equipment. In the event of a cache failure, a backup cache simply re-fetches the data from the originating store. In the case of commercial databases, there are replication schemes, journaling, and disk based backups using periodic push/update techniques and write through secondary servers. However, these approaches depend upon a fairly centralized storage system.

A traditional fault tolerant system uses N+1 devices where N devices carry the capacity and the +1 device is a in a hot standby mode. When a failure occurs in one of the N devices, the +1 device takes over but it must disrupt the system to learn the state of the device it is replacing since it cannot know the state of every possible device in the system it might have to replace. As a result, the industry has gone to a 1+1 scheme where every device has its own dedicated backup which maintains its partner's state so that failures can be seamless and not disrupt the system. However, in this scheme, half of

the devices are sitting idle and the total system requires 2N devices for implementation.

Another scheme, RAID redundancy, does not require 2N devices but uses a centralized controller. Various categories of redundancy can be configured at the controller to mirror data between storage devices and tolerate individual hardware failure. However, a failure at the controller would produce a devastating effect to the network employing this scheme. Thus, it would be desirable to provide a scheme that avoids system disruptions in the event of a failure while reducing the number of devices that sit idle during normal operation.

SUMMARY OF THE INVENTION

From the foregoing, it may be appreciated by those skilled in the art that a need has arisen for a redundancy scheme that is robust and minimizes idle devices. In accordance with the present invention, a system and method for providing redundant data load sharing in a distributed network are provided that substantially eliminate or greatly reduce disadvantages and problems associated with conventional redundancy schemes.

According to an embodiment of the present invention, there is provided a method for providing redundant data load sharing in a distributed network that includes receiving a data entry and storing the data entry in a first one and a second one of a plurality of nodes. In response to a failure of the second one of the plurality of nodes, the data entry in the failed second one of the plurality of nodes is replicated at a third one of the plurality of nodes in order to maintain data redundancy in the distributed network. Data redundancy can be retained despite node failures as long as a number of operating nodes have sufficient capacity to maintain data redundancy in the distributed system.

The present invention provides various technical advantages over conventional redundancy schemes. For example, one technical advantage is the use of all devices in the system during normal operation without requiring any idle devices to provide backup capabilities. Another technical advantage is to provide redundancy despite a failure in one of the nodes of the system. Yet another technical advantage is the ability to dynamically adjust to the number of operating and failed nodes as needed in order to maintain full

functionality of the system. Still another technical advantage is to provide an adaptive approach to providing redundancy such that only one additional device is required, more than one additional device is optional, and all devices share the load.

All devices are peers in a decentralized architecture so that a single device does not control operation in the network and a single failure would not be catastrophic for the network. The overall capacity provided by the plurality of the nodes can be dynamically adjusted up or down on demand by adding or removing nodes without reconfiguring any existing nodes or any central controller. The network of the present invention can adjust from a redundant to a non-redundant mode when the number of nodes reaches a level at which maintaining redundant copies of data is no longer possible. The network would still be in a usable state but data would not be replicated. As new nodes are added or failed nodes are restored to an acceptable operating state, the network is able to re-adjust back to a redundant mode where data can again be replicated. Other technical advantages may be readily ascertainable by those skilled in the art from the following figures, description, and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings, wherein like reference numerals represent like parts, in which:

FIGURE 1 illustrates a simplified block diagram of a distributed network;

FIGURE 2 illustrates a simplified block diagram of the distributed network during failure of a node;

FIGURE 3 illustrates a simplified block diagram of the distributed network during failure of another node;

FIGURE 4 illustrates a simplified block diagram of the distributed network during recovery of the failed node.

DETAILED DESCRIPTION OF THE INVENTION

FIGURE 1 is a block diagram of a distributed network 10. Distributed network 10 includes a plurality of nodes 12 that may be interconnected by a data network 14. Each of the plurality of nodes 12 includes a controller unit 16 to provide a decentralized controller functionality in order to supervise operation and handle movement of data entries within distributed network 10. The distribution of the controller functionality throughout distributed network 10 at each node 12 ensures that a controller functionality remains present despite a failure of a node 12 and avoids a central controller design whose failure would greatly impede network operation.

In distributed network 10, the plurality of nodes 12 provide a certain capacity to store an original data entry and a replicated data entry. For example, original data entry A is received in distributed network 10 and may be stored in node W. Node W will then copy original data entry A to generate replicated copy A´ for storage in another node, in this instance node Y. Similarly, original data entry B may be stored in Node X and replicated data entry B´ may be stored in Node W. Also, original data entry C may be stored in Node X and replicated data entry C´ may be stored in Node Z. In this manner, an original data entry and its replicated copy are stored at separate nodes. Though used in terms of original and replicated data entries for discussion purposes, there is no distinction made in distributed network 10 as to which data entry was stored first or subsequently replicated. Each data entry includes information as to where its associated data entry is redundantly stored within distributed network 10. In this manner, a first node can learn of the failure of a

second node and determine if any of its data entries had a copy stored in the failed second node and initiate replication of any such data entry at one or more other nodes according to available network capacity.

FIGURE 2 shows actions taken in distributed network 10 when a failure occurs at a node 12. For purposes of discussion, a failure occurs at Node W. Upon failure of Node W, the remaining nodes 12 receive notification as to the failure of Node W. Each remaining node 12 determines whether it is storing either an original data entry or a replicated data entry associated with data entries stored at failed Node W. If so, then the remaining nodes which have data entries associated with the data entries of failed Node W initiate replicating the appropriate data entry to a different node 12 within distributed network 10. For example, Node X determines that replicated data entry B′, associated with its original data entry B, was stored on failed Node W. Node X copies original entry B to generate re-replicated data entry B′′ and store re-replicated data entry B′′ on another working node 12. In this instance, re-replicated data entry B′′ is stored at Node Y. Re-replicated data entry B′′ may be stored on any other working node 12 pursuant to capacity restrictions. Similarly, Node Y determines that original data entry A, associated with its replicated data entry A′, was stored on failed Node W. Node Y copies replicated data entry A′ to generate re-replicated data entry A′′ for storage on any of the remaining operational nodes 12. In this instance, re-replicated data entry A′′ is stored at Node Z though it could have easily been stored at Node X.

The capacity of distributed network 10 is based on the number of nodes supported and the amount of data to

be stored. Let N be the number of nodes required to store an amount of data D. In distributed network 10, N is three nodes. The fourth, or N+1st, node is a redundant device. Though described as having one additional redundant device, distributed network 10 may have any number of additional redundant devices as desired. The additional N+1st node shares the load during normal operation with the other N nodes. The N+1st node acts as a peer instead of a standby unit and is fully functional at all times in order to enhance scalability and system performance. In this manner, no single node is idle and all nodes provide a functional capability at all times. As new data entries are received, each data entry is stored in two nodes 12 of distributed network 10. Thus, an original data entry and an associated replicated data entry are stored in different locations in distributed network 10. If a node fails, the remaining nodes 12 determine the best course of action based on the remaining physical capacity of distributed network 10. As shown above, the failure of a node 12 results in the data entries at the failed node being re-replicated across the remaining N nodes in order to retain both capacity and high availability.

FIGURE 3 shows the situation where another node fails in distributed network 10. If the minimum number of nodes N in distributed network 10 is 3, then there may be insufficient capacity to replicate the data entries of Node Z at other nodes 12 due to a failure of Node Z after the failure of Node W. In this situation, there still remains within distributed network 10 at least one copy of each data entry received in the remaining operational nodes 12. Thus, at this point no data entry has been lost as a result of a failure of a node 12. As a new

data entry is received, at least one occurrence of the new data entry will be stored in distributed network 10 but a second occurrence will most likely not be generated until the capacity of distributed network 10 is returned to its initial minimum level. There may be complex capacity decisions made that allow for replication of data entries despite the failure of a node 12 until space to do so is exhausted. Distributed network 10 may dynamically adjust capacity downward at this point to account for the failure of a node 12. Additional node failures may likewise reduce overall system capacity and actual data may be lost. However, for N being 3, distributed network 10 can survive the loss of two nodes 12 and still maintain at least one occurrence of each data entry. Distributed network 10 can survive more than two node failures if there were more than one redundant node provisioned. Any second occurrences of data entries in distributed network 10 may be subsequently replaced by newly received data entries until the initial minimum capacity is restored. The point at which redundancy can no longer be maintained is dependent upon the number of available remaining nodes and the capacity of each of the remaining nodes.

FIGURE 4 shows actions taken in distributed network 10 during recovery of Node W or the addition of a new node in distributed network 10. Node W now becomes a functional participant within distributed network 10 for sharing the load during normal operations. Previous data entries in Node W at the time of its failure are no longer considered to be present in Node W upon its recovery. As new data entries are received, they may be stored in Node W as an original data entry or a replicated data entry. For example, data entry E may be

received and stored in Node W. Node W generates replicated data entry E´ for storage in another node 12, here in Node X. Also, if the failure of Node W had forced distributed network 10 to prevent replication of data entries due to capacity limitations, there may be a data entry A´ that has no associated replicated data entry. If the recovery of Node W now provides the necessary system capacity for data replication, nodes 12 having data entries without associated replicated data entries may generate associated replicated data entries for storage at other nodes 12. For example, Node Y has data entry A´ without an associated replicated data entry. Upon determining that distributed network 10 now has sufficient capacity for replication as a result of the addition or recovery of Node W, Node Y generates replicated data entry A´´ for storage in another node 12. In the case shown, replicated data entry A´´ is stored in Node W though it could have been stored in any other node 12 according to space limitations. Similarly, data entry C in Node X may be replicated if available redundant capacity has been recovered as a result of the recovery of Node W.

As failed nodes recover, distributed network 10 may automatically determine in a distributed manner to increase capacity and resume redundant storage for high availability. Thus, upon recovery, newly received data entries may be replicated so that two occurrences are available within distributed network 10. Also, previously received data entries that were not replicated due to insufficient capacity may become replicated as capacity is recovered in distributed network 10.

A node can be any data repository which may be part of a system that performs other functions or may be

specifically intended to be used as storage. A node has a fixed capacity P and is designed to have additional storage to support replicates for its peer nodes. A network of N nodes is intended to support D discrete items of data where D is less than or equal to N times P. All nodes share in processing of the algorithm for storage and replication of data entries as equal peers. The adaptive redundancy scheme described herein may be performed in software modules distributed among nodes 12. As nodes come and go, the software modules automatically and dynamically determine courses of action to adaptively reduce capacity, maintain redundancy and availability of data, and restore capacity all based on provisioned capacity for distributed network 10 and the physical limitations at each node 12. Upon each failure and recovery, network behavior and capacity are recomputed.

Thus, it is apparent that there has been provided, in accordance with the present invention, a system and method for providing redundant data load sharing in a distributed network that satisfies the advantages set forth above. Although the present invention has been described in detail, it should be understood that various changes, substitutions, and alterations may be readily ascertainable by those skilled in the art and may be made herein without departing from the spirit and scope of the present invention as defined by the following claims. Moreover, the present invention is not intended to be limited in any way by any statement made herein that is not otherwise reflected in the appended claims.